IEEE
IEEE Signal Processing Society
IEEE Transactions on Audio, Speech and Language Processing

Enhanced Multi-domain Dialogue State Tracker with Second-order Slot Interactions

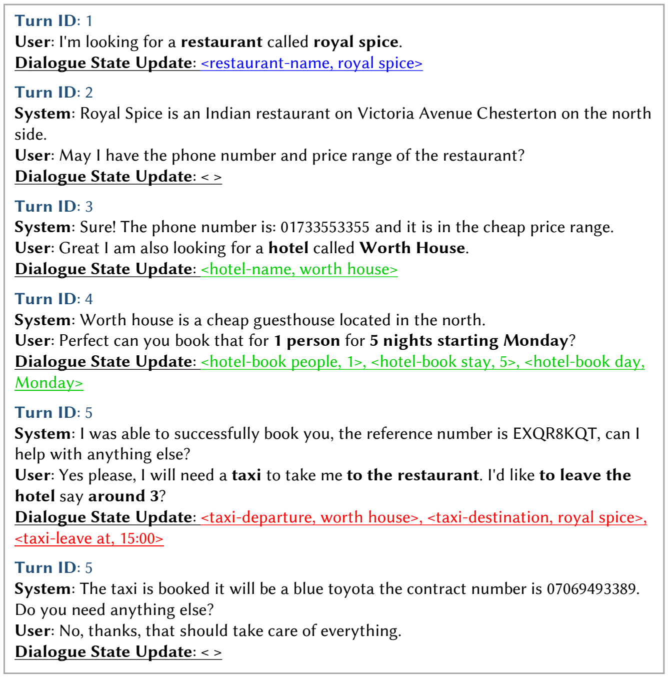
Journal:	*Transactions on Audio, Speech and Language Processing*
Manuscript ID	T-ASL-08896-2021
Manuscript Type:	Regular Paper
Date Submitted by the Author:	27-Dec-2021
Complete List of Authors:	Jiao, Fangkai; Shandong University, Department of Computer Science and Techonology Guo, Yangyang; National University of Singapore, Huang, Minlie; Tsinghua University, computer science Nie, Liqiang; Shandong University, Computer Science and Technology
Subject Category Please select at least one subject category that best reflects the scope of your manuscript:	HUMAN LANGUAGE TECHNOLOGY
EDICS:	HLT-DIAL Discourse and Dialog < HUMAN LANGUAGE TECHNOLOGY, HLT-UNDE Language Understanding and Computational Semantics < HUMAN LANGUAGE TECHNOLOGY

Enhanced Multi-domain Dialogue State Tracker with Second-order Slot Interactions

Fangkai Jiao, Yangyang Guo, Minlie Huang *Member, IEEE*, Liqiang Nie *Senior Member, IEEE*

***Abstract*—Dialogue state tracking (DST) is often used to track the system's understanding of the user goal in task-oriented dialogue systems. Existing DST methods mainly fall into two categories according to their adopted model structure: non-hierarchical and hierarchical models. The former takes the whole dialogue history as inputs during each conversation round, while the latter leverages both an utterance encoder and a dialogue encoder to efficiently model the long-term dialogue dependency. However, few of them exploit the second-order slot interaction, which refers to the pair-wise semantic relationships between different slots. As a result, these methods fall short in the context understanding throughout conversations, leading to sub-optimal performance. Towards this end, in this paper, we present a novel hierarchy-based DST framework equipped with a well-designed value copy mechanism. In particular, to model the second-order slot interaction, we firstly encode the utterance via a state reuse module to yield slot-sensitive context representation. We then selectively and effectively copy the filled values from other slots to attain more accurate state tracking. In order to evaluate the effectiveness of the proposed method, we perform extensive experiments on the widely adopted benchmark dataset MultiWOZ2.1. Our experimental results demonstrate the superiority in context understanding, as well as the strong generalization capability under a zero-shot setting compared with several DST baselines.**

***Index Terms*—Multi-domain Dialogue State Tracking, Second-order Slot Interaction, Copy Mechanism.**



Turn ID: 1
User: I'm looking for a **restaurant** called **royal spice**.
Dialogue State Update: <restaurant-name, royal spice>

Turn ID: 2
System: Royal Spice is an Indian restaurant on Victoria Avenue Chesterton on the north side.
User: May I have the phone number and price range of the restaurant?
Dialogue State Update: < >

Turn ID: 3
System: Sure! The phone number is: 01733553355 and it is in the cheap price range.
User: Great I am also looking for a **hotel** called **Worth House**.
Dialogue State Update: <hotel-name, worth house>

Turn ID: 4
System: Worth house is a cheap guesthouse located in the north.
User: Perfect can you book that for **1 person** for **5 nights starting Monday**?
Dialogue State Update: <hotel-book people, 1>, <hotel-book stay, 5>, <hotel-book day, Monday>

Turn ID: 5
System: I was able to successfully book you, the reference number is EXQR8KQT, can I help with anything else?
User: Yes please, I will need a **taxi** to take me **to the restaurant**. I'd like **to leave the hotel** say **around 3**?
Dialogue State Update: <taxi-departure, worth house>, <taxi-destination, royal spice>, <taxi-leave at, 15:00>

Turn ID: 5
System: The taxi is booked it will be a blue toyota the contract number is 07069493389. Do you need anything else?
User: No, thanks, that should take care of everything.
Dialogue State Update: < >

Fig. 1. A task-oriented dialogue instance within three domains: restaurant, hotel, taxi. The dialogue state (formatted as ⟨domain-slot, value⟩) will be tracked and updated after each conversation round. And the update rule is partially based on the related utterances, which are highlighted in bold.

I. Introduction

Dialogue State Tracking (DST) contributes an essential component in current task-oriented dialogue systems. In DST, a dialogue state is often expressed by ⟨slot, value⟩ pairs, and a state tracker is employed to estimate the user's goal in each conversation round based on the dialogue history. However, previous DST methods mainly focus on dialogue systems within a single domain, wherein the involved slots lie into one conversation topic, e.g., *restaurant reservation*. Recently, Budzianowsk et al. [1] have extended the dialogue information into multiple domains, posing new challenges to the community. Taking the dialogue history in Figure 1 as an example, it describes three different domains: *restaurant*, *hotel*, and *taxi*. The target user starts the conversation with *restaurant* reservation and ends it with *taxi* booking after requesting information for a *hotel*. In view of this, it is necessary for the DST models to determine the values for the given slot based on its attached domain at each turn, while some of them even demands inference from multi-turns and cross-domains [2].

Traditional dialogue state trackers rely heavily on hand-crafted features and domain-specific lexicons [3]–[5]. With the prosperity of deep learning era, neural models have made considerable progresses in recent years [6]–[8]. In general, these neural methods can be roughly categorized into two groups in terms of the model architecture: non-hierarchy-based and hierarchy-based DSTs. Methods from the first group consider the whole dialogue history as inputs during each conversation round [2], [9]–[12]. Nevertheless, under the multi-domain setting, the dialogue often lasts more rounds and hence produces a longer context. In a sense, it is difficult to model the long-term dialogue dependency owing to gradient vanishing (RNN-based models [13]) or fixed context length (Transformer-based models [14]) for these non-hierarchical approaches. Besides, the dialogue history requires repeated computation when consuming the recent coming turns, leading to largely increased overhead. By contrast, the hierarchy-based

Fangkai Jiao is with the Department of Computer Science and Technology, Shandong University, Shandong, China, (e-mail: jiaofangkai@hotmail.com).

Yangyang Guo is with the School of Computing, National University of Singapore, (e-mail: guoyang.eric@gmail.com).

Minlie Huang is with the Department of Computer Science, Tsinghua University, Beijing 100084, China, (e-mail: aihuang@tsinghua.edu.cn).

Liqiang Nie is with the Department of Computer Science and Technology, Shandong University, Shandong, China, (e-mail: nieliqiang@gmail.com).

ones are designed to alleviate these limitations by compressing the representation of the utterance in each dialogue turn [15]–[17]. Specifically, these models usually employ an utterance encoder and a dialogue encoder simultaneously to track dialogue states. For example, SUMBT [17] leverages a BERT encoder and a Multi-Head Attention [18] module to generate the representation of the utterance and the slot, respectively. And a Gated Recurrent Unit (GRU) is then adopted to manage the dialogue encoding with respect to each slot.

Though the aforementioned methods have gained superiority over their contenders, modeling the second-order slot interactions is largely unexplored. The second-order slot interaction we refer to here, implies the pair-wise relationship among slots, which accounts for predominant effect on estimating user's goal in the dialogue. For example, when tracking the state for slot *taxi-destination* in Figure 1 pertaining to the fifth turn, the entity *restaurant* is first located. Afterwards, the DST model should traverse the dialogue history for the name of the restaurant, which is already circled by the slot *restaurant-name* in the first turn. The same observation can also be found for slots *taxi-departure* and *restaurant-name*. However, existing DST models regard each slot as an independent query, namely, the first-order slot interaction, and thus fail to capture these second-order relationships, resulting in sub-optimal performance. In order to approach this problem, we propose to model the second-order slot interactions for DST to capture the long-term dialogue dependency as well as enhance the context understanding.

By consolidating this idea for DST, in this paper, we develop a novel hierarchical DST framework equipped with a delicately devised copy mechanism. In specific, to encode the utterance and slot inputs, we firstly adopt a hierarchical structure based on BERT [19] for better capturing the long-term dialogue dependency, where a state reuse mechanism is employed to avoid heavy computation in non-hierarchical models. Thereafter, we leverage a Multi-Head Attention [18] model as the belief tracker for managing the dialogue states. Finally, we design a value copy mechanism to effectively model the second-order slot interactions. Our copy mechanism is composed of three modules: value attention, slot attention, and dynamic gate unit. In particular, the value attention is designed for slot filling based on the state tracker from previous turns. The slot attention module aims to determine the copy path from source slots to the target slot according to the slot-specific contextual representation. And the dynamic gate unit controls how much information from the summarized values have being fused for the final value prediction.

To validate the effectiveness of the proposed method, we conduct extensive experiments on the most widely exploited benchmark dataset - MultiWOZ2.1, and achieve superior performance, i.e., 53.65% joint accuracy. Furthermore, we justify the viability of our method under the zero-shot setting, whereby the knowledge from certain domains is eliminated.

In summary, our contribution of this paper is three-fold:

- We devise a new value copy mechanism to model the second-order slot interactions in DST towards enhancing the context understanding. In addition, our method achieves superior results on the benchmark dataset over various strong baselines.
- We present a novel hierarchical framework based on BERT to capture the long-term dialogue dependency for DST.
- We conduct extensive experiments on the MultiWOZ2.1 dataset to verify the effectiveness of the proposed method. Moreover, we have open-sourced our code and experimental settings to facilitate other researchers[1].

The remaining of this paper is organized as follows. The related work is elaborated in Section II, followed by the details of the proposed method in Section III. We then present the experimentation in Section IV, and demonstrate the obtained results in Section V. We finally conclude this paper and discuss the future directions in Section VI.

II. Related Work

In this section, we mainly discussed two directions of studies which are closely related to this work: dialogue state tracking and the copy mechanism.

A. Dialogue State Tracking

Dialogue state tracking has attracted much attention over the past few years. On the basis of the model architecture, the existing DST methods can be classified into two categories: the non-hierarchical and the hierarchical models. In fact, the methods in the former category take the full dialogue history as inputs during each conversation round [2], [9], [20], [21]. Benefiting from machine reading comprehension [22], [23] and pre-trained language models [19], [24], [25], these non-hierarchical methods have achieved favorable results. However, they often suffer from the heavy computation problem due to the repetitive encoding of dialogue history. Besides, it is hard for these methods to capture the long-term dependency. To cope with these limitations, hierarchical models in the latter are developed to reuse the hidden states of each utterance and efficiently capture longer context [13]. Concretely, both an utterance encoder and a dialogue encoder are employed for better state tracking.

We recognize one defect of the existing methods, that most of them take each slot as an independent query, leaving the slot interaction untouched. For instance, SUMBT [17] converts each slot as independent questions to extract the corresponding value information from the dialogue history, and some generative methods take the slot as specific condition for generating the corresponding dialogue states [2], [26], [27]. In the light of this, some pipeline models [28], [29] take the parsed dialogue states till the last turn as extra inputs to model the interactions between different slots, whereas the methods may be easily damaged by the error propagation problem. Besides, some works employ the manually constructed graph [30] or semantic similarity matrix [31] to capture the shallow dependencies between different slots, which can be difficult to model the potential correlations among slots completely. In this paper, we

[1] https://github.com/SparkJiao/dst-multi-woz-2.1/tree/master/SUMBT.

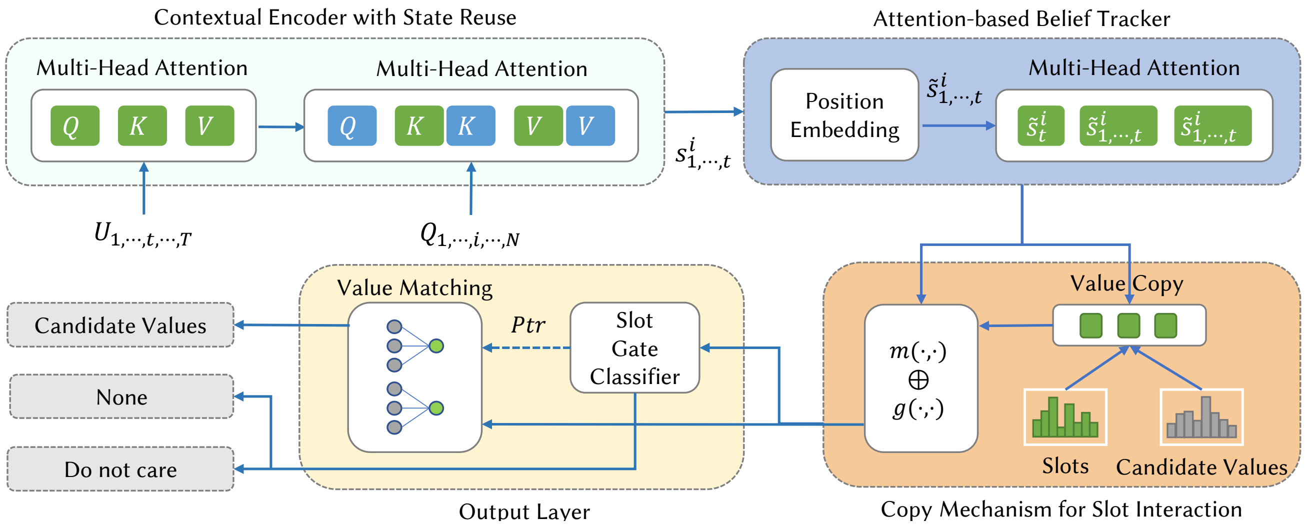

Contextual Encoder with State Reuse
Multi-Head Attention
Q K V
Multi-Head Attention
Q K K V V
$U_{1,\cdots,t,\cdots,T}$
$Q_{1,\cdots,i,\cdots,N}$
$s^i_{1,\cdots,t}$
Attention-based Belief Tracker
Position Embedding
$\tilde{s}^i_{1,\cdots,t}$
Multi-Head Attention
$\tilde{s}^i_t$ $\tilde{s}^i_{1,\cdots,t}$ $\tilde{s}^i_{1,\cdots,t}$
Value Copy
$m(\cdot,\cdot)$ $\oplus$ $g(\cdot,\cdot)$
Slots
Candidate Values
Copy Mechanism for Slot Interaction
Value Matching
Ptr
Slot Gate Classifier
Candidate Values
None
Do not care
Output Layer

Fig. 2. Overall framework of the proposed CP-DST model. It firstly takes as inputs the dialogue (U) and slots (Q) with the Contextual Encoder, and then generates the slots-specific contextual representation via the Belief Tracker. In the next, the copy mechanism is developed to transfer values from previous turns for the target slot. Finally, the Output Layer yields the prediction based on the Slot Gate Classifier as well as the Value Matching modules.

propose to use a more intuitive method to model the second-order slot interactions by transforming it to a binary problem, i.e., whether some values can be copied from other filled slots. In particular, we introduce a slot attention mechanism to reflect the deep interactions among slots and decide the copy path, followed by a value attention module to summarize the state representations from previous turns.

B. Copy Mechanism

The copy mechanism is extensively studied in text generation to solve the out-of-vocabulary (OOV) problem. Pointer-network [32] is firstly proposed to produce a distribution by adopting the attention mechanism upon the tokens from the input sequence, which is then leveraged to yield the output sequence. Inspired by this, several studies have extended the pointer-network via a hybrid fashion of both *generation* and *copy*. Concretely, the words in the input sequence are re-weighted and selectively copied to generate the output sequence. This strategy has been proven effective in text summarization [33], question answering [34], relation fact extraction [35] and query suggestion [36] tasks, to name a few. Beyond the application in the natural language processing field, researchers have also exploited the copy mechanism in cross-modal problems. For example, LSTM-C [37] employs a novel LSTM-based framework augmented with the copy mechanism to tackle the few-shot learning task in image captioning. In particular, the model generates each word in the outputted caption based upon two sets: the paired image-sentence set and the constructed unpaired object recognition set. In this way, the object words scarcely seen in the training set can be generated by copying the object class in the object recognition set for each image.

In this paper, we endeavor to adapt the copy mechanism for the multi-domain DST, as some state values can be directly copied from other slots.

III. Proposed Method

Given a dialogue $\mathcal{D} = \{U_1^s, U_1^u, U_2^s, U_2^u, \cdots, U_T^s, U_T^u\}$, all slots in ontology $\mathcal{S} = \{S_1, S_2, \cdots, S_N\}$ and the corresponding candidate values $\mathcal{V}^i = \{V_1^i, V_2^i, \cdots, V_K^i\}$ for slot S_i, where U_t^s, U_t^u are the token sequences of the system and user utterance at turn t, respectively, the goal of DST is to correctly predict the value for each slot at all dialogue turns.

A typical resolution of DST is comprised of two process steps [2], [17]: a tentative prediction of whether the current slot can be filled with *none* or *don't care* is firstly given; if not, the model in the second move will select the most plausible value from the candidate value set. When it refers to the latter step, existing methods often score each value candidate based on the context-aware slot representations. However, most of them have not taken the second-order slot interaction into consideration, namely, the contextual information across slots have not been well exploited. To effectively leverage this auxiliary information, in this paper, we propose a novel DST method, dubbed as CP-DST, to selectively weight and copy the values from other slots for the target one.

The framework of CP-DST is shown in Figure 2. Given the dialogue history, the utterance and slots are jointly encoded by the contextual encoder equipped with a state reuse mechanism. And then the attention-based belief tracker takes the encoded slot representation as input to obtain the slot-specific context. Thereafter, a soft copy mechanism is devised to selectively and effectively copy possible values from other slots for the final value prediction. In what follows, we will detail these three parts in our model sequentially.

A. Contextual Encoder with State Reuse

As BERT [19] has shown strong generalization capability in various NLP tasks, we thus resort to the pre-trained BERT model to generate the deep bidirectional contextual representations for utterance, slot and value encoding.

Utterance Encoder. To encode the utterance information into our model, we formulate the input sequence for accommodating BERT as,

$$U_t = [CLS] \oplus U_t^u \oplus [SEP] \oplus U_t^s \oplus [SEP],$$

where $\oplus$ denotes the concatenation operation, $[CLS]$ and $[SEP]$ are the classification and separation token in BERT, respectively. We then use the BERT model to obtain the utterance representation, where the Multi-Head Attention [18] is adopted. For concise expression, throughout this paper, we refer to the Multi-Head Attention as $\mathrm{MHA}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are query, key and value matrices, respectively. In this way, the process in a single layer of BERT can be represented as,

$$\mathbf{O}_l^U = \mathrm{MHA}(\mathbf{H}_{l-1}^U, \mathbf{H}_{l-1}^U, \mathbf{H}_{l-1}^U),$$
$$\mathbf{H}_l^U = \mathrm{FeedForward}(\mathbf{O}_l^U),$$

where l denotes the l-th layer and the conversation turn index t is ignored, and $\mathbf{H}_0^U$ denotes the output of the embedding layer in BERT. For simplicity, we do not express the layer normalization and residual connection as in the original BERT model.

Slot Encoder. Most of the existing DST methods [27], [29] generate specific utterance representation for each slot, causing heavy computation due to the extremely long lengths of the dialogue. To address this issue, we adopt a state reuse mechanism [14] to restrict the direction of the attention flow to be utterance-to-slot so that the repetitive encoding of utterance can be avoided. In particular, the input sequence of each slot is formulated as,

$$Q_i = [CLS] \oplus S_i \oplus [SEP].$$

And then we extend the MHA module for the i-th slot S_i as,

$$\mathbf{O}_l^{S_i} = \mathrm{MHA}(\mathbf{H}_{l-1}^{S_i}, [\mathbf{H}_{l-1}^U, \mathbf{H}_{l-1}^{S_i}], [\mathbf{H}_{l-1}^U, \mathbf{H}_{l-1}^{S_i}]),$$
$$\mathbf{H}_l^{S_i} = \mathrm{FeedForward}(\mathbf{O}_l^{S_i}),$$

where $[,]$ signifies the row-wise concatenation.

At last, we leverage the encoded vector $\mathbf{s}^i$ corresponding to the special [CLS] token in Q_i for further processing.

Value Encoder. For the encoding of the candidate values, we use the vector generated by the mean pooling of the output from another BERT encoder with all parameters being fixed,

$$\mathbf{v}_j^i = \mathrm{MeanPooling}(\mathrm{BERT}_{\mathrm{fix}}([CLS] \oplus V_j^i \oplus [SEP])).$$

B. Attention-based Belief Tracker

To manage the dialogue states as the conversation progresses, the belief tracker should be employed to update the states for all slots with the corresponding hidden states of each round. Different from SUMBT [17], we leverage the Masked Multi-Head Self-Attention [18] as the belief tracker, where the inputs after current turn will be masked. We do not apply GRU [38] due to the gradient vanishing problem. This requires us to add an extra position embedding to the dialogue representation. Therefore the input to the belief tracker can be formulated as,

$$\tilde{\mathbf{s}}_t^i = \mathbf{s}_t^i + \mathbf{e}_t,$$

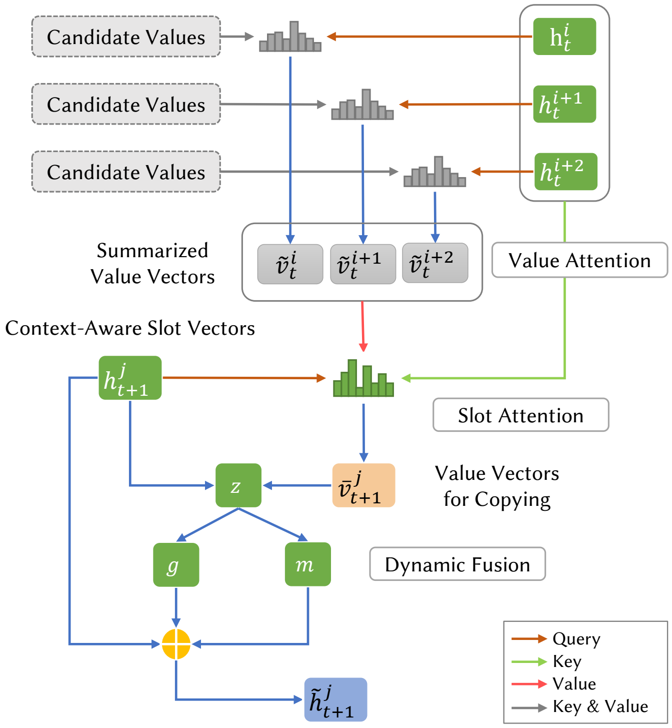


Fig. 3. Overview of our Copy Mechanism.

where $\mathbf{e}_t$ is the position embedding at the t-th position, which shares weights with that of the BERT encoder. And then the slot-specific context hidden state can be obtained through,

$$\mathbf{h}_t^i = \mathrm{MHA}(\tilde{\mathbf{s}}_t^i, [\tilde{\mathbf{s}}_0^i, \tilde{\mathbf{s}}_1^i, \cdots, \tilde{\mathbf{s}}_t^i], [\tilde{\mathbf{s}}_0^i, \tilde{\mathbf{s}}_1^i, \cdots, \tilde{\mathbf{s}}_t^i]). \quad (1)$$

C. Copy Mechanism for Slot Interaction

Given the slot-specific contextual hidden state $\mathbf{h}_t^i$ from the belief tracker, a *Copy Mechanism* is devised to selectively copy possible values from other slots based on the corresponding slot interactions. Specifically, for the representation of each slot $\mathbf{h}_t^i$, we consider the second-order interaction as the relevance modeling between $\mathbf{h}_t^i$ and $\mathbf{h}_{t-1}^j$, where $j \in [1, N]$. Afterwards, the copy mechanism is employed to summarize the hidden states of possible copied values according to the relevance for further value matching and classification. Towards this end, we first use a value attention module to determine the filled value representation in previous turns and then a slot attention module is designed to specify the copy path from the source slot to the target one. Given that not all slots require to copy values for accurate prediction, and thence, a dynamic gate unit is used to control how much information will be fused into the slot representation. A detailed illustration can be found in Figure 3.

In the following, we adopt a simple attention function proposed by [23] for both the value and slot attention, which has been proven effective in machine reading comprehension,

$$\mathrm{Att}(\mathbf{a}, \mathbf{b}) = f(\mathbf{W}\mathbf{a})^T \mathbf{D} f(\mathbf{W}\mathbf{b}),$$

where $f(\mathbf{x})$ is an activation function and we use $f(\mathbf{x}) = \mathrm{ReLU}(\mathbf{x})$ in our implementation, and $\mathbf{D}$ is a diagonal matrix.

Both $\mathbf{W}$ and $\mathbf{D}$ are trainable matrices with the same dimension as the input vectors $\mathbf{a}$ and $\mathbf{b}$.

Value Attention. To avoid error propagation from the discrete dialogue states predicted previously, we leverage the attention mechanism to obtain the weighted sum of all candidate value embeddings with respect to the slot-specific context hidden state. The relevance scores between each slot and the corresponding candidate value is attained by,

$$\mathbf{M}_{ijt} = \mathrm{Att}(\mathbf{h}_t^i, \mathbf{v}_j^i). \tag{2}$$

In this way, the attended value embedding for each slot becomes:

$$\tilde{\mathbf{v}}_t^i = \sum_j \alpha_{j,t}^i \mathbf{v}_j^i, \;\; \alpha_{j,t}^i \propto \exp(\mathbf{M}_{ijt}).$$

Besides, we use a cross-entropy loss restriction here to achieve more accurate attention computation,

$$\mathcal{L}_v = -\sum_t^T \sum_i^N \log \mathbf{P}_{i,:,t}^v \mathbf{y}_{i,t}^v, \;\; \mathbf{P}_{i,j,t}^v \propto \exp(\mathbf{M}_{ijt}), \tag{3}$$

where $\mathbf{y}_{i,t}^v$ is the ground-truth one-hot value index label.

Slot Attention. To model the second-order slot interaction, we employ another attention module to determine the relevance scores between the plausible source slot and target one. Different from the previous graph-based model [29], we simply employ $\mathbf{h}_{t-1}^j$ as the representation of source slot to enrich the contextual information. In formulation, the relevance scores are obtained through,

$$\mathbf{N}_{ijt} = \mathrm{Att}(\mathbf{h}_t^i, \mathbf{h}_{t-1}^j). \tag{4}$$

And then the summarized value representation from all source slots can be obtained via a weighted sum followed by a fully connected layer,

$$\begin{cases} u_t^i = \sum_j \beta_{j,t}^i \tilde{\mathbf{v}}_{t-1}^j, \quad \beta_{j,t}^i \propto \exp(\mathbf{N}_{ijt}), & (5) \\ \bar{\mathbf{v}}_t^i = \mathrm{gelu}(\mathbf{W}_u \mathbf{u}_t^i + \mathbf{b}_u), & (6) \end{cases}$$

Dynamic Fusion. Finally, we employ an effective function to fuse the copied value hidden states and initial user slot states,

$$\begin{cases} m(\mathbf{h}_t^i, \bar{\mathbf{v}}_t^i) = \tanh(\mathbf{W}_m \mathbf{z}_t^i + \mathbf{b}_m), & (7) \\ g(\mathbf{h}_t^i, \bar{\mathbf{v}}_t^i) = \mathrm{sigmoid}(\mathbf{W}_g[\mathbf{z}_t^i] + \mathbf{b}_g), & (8) \\ \mathbf{z}_t^i = [\mathbf{h}_t^i; \bar{\mathbf{v}}_t^i; \mathbf{h}_t^i - \bar{\mathbf{v}}_t^i; \mathbf{h}_t^i \circ \bar{\mathbf{v}}_t^i], & (9) \end{cases}$$

where $\mathbf{W}_m \in \mathcal{R}^{4d \times d}$, $\mathbf{W}_g \in \mathcal{R}^{4d \times 1}$, $m(\mathbf{x}, \mathbf{y})$ is the fused hidden state and $g(\mathbf{x}, \mathbf{y})$ is the gate to control how much information of copied value should flow to the target state. The fused hidden representation is formulated as,

$$\begin{aligned} \tilde{\mathbf{h}}_t^i &= \mathrm{Fuse}(\mathbf{h}_t^i, \bar{\mathbf{v}}_t^i) \\ &= g(\mathbf{h}_t^i, \bar{\mathbf{v}}_t^i) \circ m(\mathbf{h}_t^i, \bar{\mathbf{v}}_t^i) + (1 - g(\mathbf{h}_t^i, \bar{\mathbf{v}}_t^i)) \circ \mathbf{h}_t^i, \end{aligned} \tag{10}$$

where $\circ$ denotes the element-wise multiplication. Given that the slot-specific hidden states in each round have been updated, we need to manage the dialogue states across conversation again. Therefore, another belief tracker layer is used to track the updated hidden states and the output is reduced to $\bar{\mathbf{h}}_t^i$.

D. Training Strategy

Slot Gate Classification. For each slot, we firstly map the user state vector $\bar{\mathbf{h}}_t^i$ to a probability distribution over *ptr*, *none* and *do not care* classes. If the classifier predicts *none* or *do not care*, we fill the slot with “none” or “do not care” directly. Otherwise, the value matching network is employed to predict a value based on the ontology. The slot gate classifier is defined as,

$$\mathbf{P}_{i,t}^G = \mathrm{softmax}(\mathbf{W}_G \bar{\mathbf{h}}_t^i + \mathbf{b}_G) \in \mathcal{R}^3. \tag{11}$$

Value Matching. For calculating the relevance score between the embedding of candidate values and user states of each slot, a bi-linear function is adopted to measure the slot-value correlation followed by a softmax function to obtain the probability distribution over candidate values:

$$\mathbf{P}_{i,j,t}^{ptr} = \mathrm{softmax}((\bar{\mathbf{h}}_t^i)^T \mathbf{W}_p \mathbf{v}_j^i). \tag{12}$$

Optimization. In particular, the slot gate classification and value matching are supervised via the cross entropy. As the former one, the loss function is formulated as,

$$\mathcal{L}_c = -\sum_t^T \sum_i^N \log \mathbf{P}_{i,t}^G \mathbf{y}_{i,t}^G, \tag{13}$$

where $\mathbf{y}_{i,t}^G$ is the one-hot slot gate label. And for the latter one, the loss function is defined as,

$$\mathcal{L}_m = -\sum_t^T \sum_i^N \log \mathbf{P}_{i,:,t}^{ptr} \mathbf{y}_{i,t}^v, \tag{14}$$

where $\mathbf{y}_{i,t}^v$ is the ground-truth one-hot value index label. Therefore the ultimate objective function is given as,

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_m + \lambda \cdot \mathcal{L}_v, \tag{15}$$

where $\mathcal{L}_v$ is the cross-entropy loss supervised for the value attention and λ is a hyper-parameter which is determined by the validation set.

As to the inference stage, we first classify the slot gate with the highest probability. And if the output is *none* or *do not care*, we directly fill it in the corresponding slot. Otherwise, we select the value from the candidate values with the highest matching score as the final target prediction.

IV. Experimental Setup

A. Dataset

We evaluated the effectiveness of our method on the benchmark dataset - MultiWOZ2.1 [39], which is collected from a *Wizard of Oz* style test and falls into *seven* distinct domains. Since the quantity of the *hospital* and *police* domains is quite sparse and these two domains only exist in the training set, we therefore removed the related instances following [2]. As a result, there are in total 30 slots and over 10,000 dialogues in the resultant dataset. Moreover, we used the generic train/test splits provided in the original dataset. To ensure a fair comparison with [40], we traversed the whole dataset to construct the ontology so that all values can be found in the candidate list. Same training ontology is leveraged for other ontology-based models such as SUMBT [17]. Table I shows the statistics of the dataset.

TABLE I
STATISTICS OF THE MULTIWOZ2.1 DATASET. THE SYMBOL #D- DENOTES THE NUMBER OF DIALOGUES.

Domain	Taxi	Hotel	Train	Attraction	Restaurant	All
#D-Train	3,813	3,103	1,654	3,381	2,717	8,438
#D-Valid	438	484	207	416	401	1,000
#D-Test	437	494	195	394	395	1,000
Slot	arrive by departure destination leave at	area, internet, name parking, price range stars, type, book day book people, book stay	arrive by book people day, departure destination, leave at	area name type	area, food, name book day, book time book people price range	-

B. Implementation Details

We built our model based on SUMBT[2] [17]. We fixed the number of attention heads used in the belief tracker to be 12. For the concatenated utterances, slots and values, their max lengths are set to be 108, 6 and 20, respectively. The coefficient of the value attention supervision is fixed 1.0. Besides, we employed the pre-trained BERT-base-uncased models as our contextual encoder and candidate value encoder, respectively. For other parameters, the default initialization method in Pytorch[3] is utilized to initialize the corresponding weights. And the BertAdam is adopted as the model optimizer, with the peak learning rate being 4e-5 and the warm-up proportion being 0.1. We set the dropout probability to 0.1 and the batch size is fixed to be 1 for a fast convergence and the model is trained for at most 5 epochs. During training, we simultaneously evaluated the model on the development set per 2,000 steps, and the model with the highest joint accuracy is saved for prediction in the testing set.

C. Compared Baselines

We compared the performance of our model with several strong baselines. A short introduction of these methods is provided in the following.

- **HyST** [41] employs a hierarchical encoder and uses a pre-defined ontology setting.
- **DST Reader** [42] reformulates the DST task as a reading comprehension task. The prediction of each slot is a span over tokens within the dialogue history. The model follows an attention-based neural network architecture and combines a slot carryover prediction module and slot type prediction module.
- **SUMBT** [17][4] exploits a hierarchical structure with BERT as the encoder for utterance and a GRU as the belief tracker. After encoding, it scores every candidate slot-value pairs in a non-parametric manner using the Euclidean distance.
- **TRADE** [2] adopts a Seq2Seq structure to encode the dialogue context and decode the corresponding value for each given domain and slot.
- **NADST** [9] uses the Seq2Seq structure with a non-auto-regressive decoder to generate the dialogue state sequence during each conversation round at once.
- **MA-DST** [27] employs a similar model structure with TRADE and extends it with a cross-attention layer and a self-attention layer.
- **DSTQA** [29] takes the DST task as a question-answering problem and adopts a non-hierarchical model structure similar to [22] with span extraction and value matching. Moreover, it uses a graph network to model the relationships between slots.
- **SOM-DST** [28] employs a Seq2Seq model with BERT as the encoder, which takes the dialogue history and previous dialogue state as inputs. It selectively overwrites the dialogue states with the pointer-generator network.

D. Evaluation Metric

The Joint Accuracy (**Joint Acc.**) is adopted for all the experiments. It compares the predicted dialogue states with the ground truth ones, and the output at each dialogue turn is deemed to be correct (i.e., 1.0 accuracy) if and only if the predicted values exactly match the ground truth ones in all slots. And then the joint accuracy is averaged over the number of dialogue turns.

V. EXPERIMENTAL RESULTS

In order to validate the effectiveness of our proposed method, we conducted extensive experiments under the qualitative and quantitative settings. In particular, the experiments mainly aim to answer the following research questions:

- **Q1**: Can our proposed method outperform the previous state-of-the-art DST methods in terms of the Joint Acc.?
- **Q2**: Why the second-order slot interaction is necessary in our method?
- **Q3**: How does the hyper-parameter, i.e., λ in Equation 15, affect the final model performance?
- **Q4**: Why the devised copy mechanism is effective and what kinds of mistakes will the proposed method make?

A. Overall Performance (Q1)

Table II shows the joint accuracy of our model with other baselines on the test set of MultiWOZ2.1. For our model, we reported results under two settings, the model with and without the copy mechanism. The observations are as follows:

- From the table, we observe that CP-DST surpasses all the baselines consistently with significant improvements. Besides, compared with the variant CP-DST w/o CP, CP-DST gains a 1.14% absolute improvement, indicating the

[2] https://github.com/SKTBrain/SUMBT.

[3] pytorch.org.

[4] We retrained SUMBT using the same ontology with other models for a fair comparison.

TABLE II
JOINT ACCURACY ON THE TEST SET OF MULTIWOZ2.1. CP-DST W/O CP DENOTES OUR MODEL WITHOUT THE COPY MECHANISM. BERT REPRESENTS THE ENCODER ARE INITIALIZED WITH BERT-BASE, AND THE PRE-DEFINED ONTOLOGY IS PROVIDED BY THE EVALUATED DATASET. *: RESULTS FROM BY OUR EXPERIMENTS. †: RESULTS REPORTED BY [39]. BEST RESULTS ARE HIGHLIGHTED IN BOLD.

Model	BERT	Pre-defined Ontology	Joint Acc.
DST Reader†			36.40
HyST		✓	38.10
TRADE			45.60
SUMBT*	✓	✓	48.13
NADST			49.04
MA-DST			51.04
DSTQA		✓	51.17
SOM-DST	✓		53.01
CP-DST w/o CP	✓	✓	52.51
CP-DST	✓	✓	**53.65**

- viability of modeling the second-order slot interaction and the effectiveness of the devised copy mechanism.
- The methods employing BERT as the backbone usually yield superior results than others. This validates the generalization and effectiveness of the BERT framework and the pre-trained BERT model. One exception is SUMBT, which encodes the utterance and slot sequence separately with the parameters of the slot encoder being fixed. It thus produces much inferior performance than other models which are fine-tuned based on BERT.
- The methods using the pre-defined ontology mostly achieve better accuracy than those without. The reason for this is that models leveraging the pre-defined ontology can largely reduce and simplify the task compared with those using an open vocabulary.

TABLE III
MODEL PERFORMANCE COMPARISON UNDER THE ZERO-SHOT SETTING WHERE THE SELECTED DOMAIN IS ELIMINATED IN THE TRAINING SET. W/O CP DENOTES OUR METHOD WITHOUT CONSIDERING THE SECOND-ORDER SLOT INTERACTION.

Model	Attraction	Hotel	Restaurant	Taxi	Train
TRADE	20.06	14.20	12.59	59.21	22.39
MA-DST	22.46	16.28	13.56	59.27	22.76
CP-DST	**30.93**	**18.17**	**16.672**	**60.77**	**22.85**

B. Zero-shot Setting Comparison (Q2)

We compared the performance of our method with several baselines under a zero-shot setting, where the selected domain is removed from the training set and test set contains only the dialogues from the target domain. From the results shown in Table III, we see that our methods outperform the baselines significantly on domains of *attraction*, *hotel* and *restaurant*, demonstrating the generalization capability of our proposed method.

C. Incomplete Slot Interaction Analysis (Q2)

During training, we selected three domains, *hotel*, *restaurant* and *taxi*, and masked the attention scores from the specific slots in each of these domains to test the performance with incomplete slot interactions. As shown in Table IV, the performance drops when the connections between specific slots are removed. Specifically, the model obtains the worst result without the slot interaction in the *taxi* domain. The reason for this is that the slots in *taxi* domain share the most common candidate values with slots in other domains.

TABLE IV
MODEL PERFORMANCE OF INCOMPLETE SLOT INTERACTIONS.

Masked Domain	None	Hotel	Restaurant	Taxi
Joint Acc.	53.65	53.025	52.455	52.252

TABLE V
PERFORMANCE COMPARISON WITH AND WITHOUT CONSIDERING THE CONTEXTUAL INFORMATION OF OUR METHOD ON MULTIWOZ2.1.

Model	Joint Acc. (Dev)	Joint Acc. (Test)
CP-DST	57.36	53.65
CP-DST w/o CP	56.71	52.51
CP-DST w/. embedding	56.41	52.71

D. Context Understanding (Q2)

DSTQA [29] employs the context-independent slot vectors and its corresponding value embedding as the graph node representation, which cannot fully exploit the context information. In the light of this, we attributed part of the improvement of our method to the devised attention mechanism, where both the query $\mathbf{A}$ and the key $\mathbf{B}$ in Equation 4 are relevant to the slot-specific context information. Hence, we conducted experiments to verify its effectiveness. Specifically, we redefined $\mathbf{N}_{ijt}$ in Equation 4 as,

$$\mathbf{N}_{ijt} = \mathrm{Att}(\mathbf{h}_t^i, \mathbf{c}^{S^i} + \mathbf{u}_t^i),$$

where

$$\mathbf{c}^{S^i} = \mathbf{C}_{[\mathbf{CLS}]}^{S^i}, \;\; \mathbf{C}^{S^i} = \mathrm{BERT}(Q_i),$$

where $\mathbf{c}^{S^i}$ is independent from the utterance U_t. We compared the performance of this variant using partial context information with our model and illustrated the results in Table V. We found that this variant shares similar performance with the one without using the proposed copy mechanism but is less favorable to the final model CP-DST. This demonstrates the necessity of incorporating the slot-specific contextual information of dialogue into the slot interaction modeling.

E. Supervision for Value Attention (Q3)

We used λ to supervise the value attention in Equation 2. Figure 5 shows the effect of the value attention loss with respect to λ. With the magnitude of λ spanning from 0 to 1, the model performance drops first, and then increases when λ is near 1. One possible reason for this case is that the value attention module can grasp more relevant values when λ is small.

F. Case Study (Q4)

We illustrated two successful cases from our method in Figure 4. There are three important kinds of outputs from our

System: ... . Now, what type of accommodations are you looking for today? **User**: The hotel should be in the same area as the restaurant and should include free wifi. **Dialog States:** (hotel-area, **north**) **Gate: 0.5146** **Slot Attention: <restaurant-area, 1.0>** **Value Attention in Last Round: <north, 1.0>**	**System**: It's 01223304705. Do you need anything else? **User**: Yeah, I need a restaurant. They need to serve Indian food and be in the same area as Funky Fun House. **Dialog States:** (restaurant-area, **east**) **Gate: 0.4294** **Slot Attention: <attraction-area, 1.0>** **Value Attention in Last Round: <east, 1.0>**

(a) The 4-th turn of Dialogue PMUL4643.

(b) The 3-th turn of Dialogue PMUL3336.

Fig. 4. Case study for the three modules in our copy mechanism: slot attention, value attention and the dynamic gate unit. The state of the attention mechanism is formulated as ⟨ attended item, attended score ⟩ pairs. The item with the highest attention score is given.

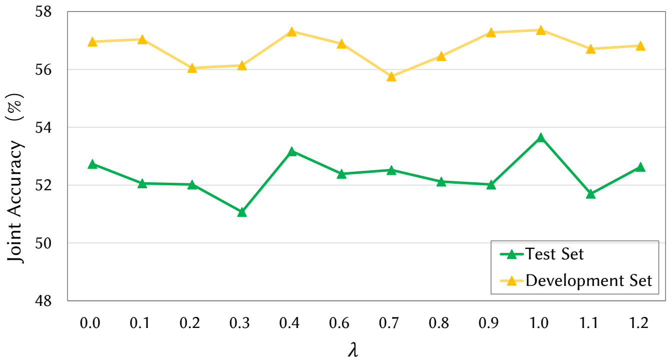


Fig. 5. The model performance with respect to different λ for the value attention loss.

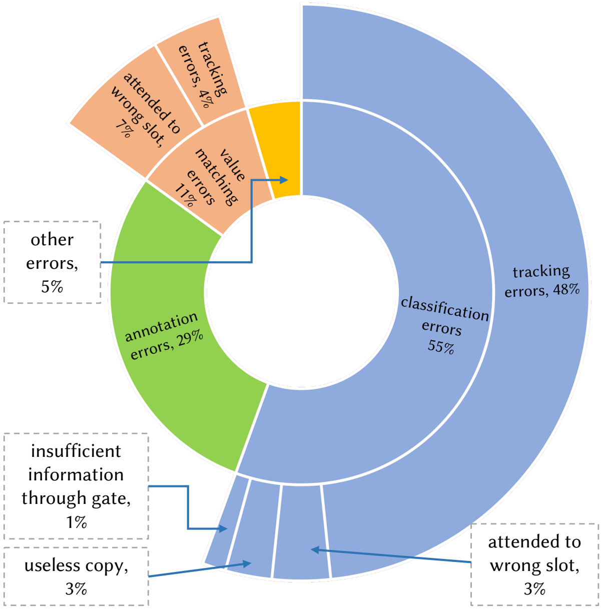


Fig. 6. Error types of our method over the MultiWOZ2.1 dataset.

CP mechanism: the value attention score, the slot attention score, and the gate unit score, which are defined in Equation 2, 4 and 8, respectively. For simplicity, we only demonstrated the attended items with the highest attention score. The first two outputs describe how accurately the attended item and the gate score reflects how much information is fused for prediction. As shown in Figure 4(a), the user asks the agent to book a hotel located at the same area with the previously booked restaurant. Surprisingly, the attention scores in both slot attention and value attention are 1.0, which exactly matches to the user's goal for *north* in the *Value Attention* and *restaurant-area* in the *Slot Attention*, respectively. Besides, the gate score reaches above 0.5, indicating half of the information for predicting *hotel-area* is exactly extracted from *restaurant-area*. The same observation can also be found in Figure 4(b), where the copy path becomes *east* $\mapsto$ *attraction-area* $\mapsto$ *restaurant-area*.

G. Error Analysis (Q4)

Figure 6 shows the different types of model prediction errors on the MultiWOZ2.1 dataset made by CP-DST. At the first glance, the *annotation errors* and *classification errors* account for 29% and 55% of the overall errors, respectively. The *annotation errors* represent that the predictions are incorrect because the ground truth values are mislabelled, which is extremely hard to avoid. The *classification errors* are pertaining to the dialogue states which are not well classified, i.e., the slots should be filled with *ptr* or *do not care* but predicted as *none* instead. The *value matching errors* are produced by a wrong value candidate selection yet the slot gate is correctly classified. Compared with the *classification errors*, they are much less frequent (11% vs 55%) since the categories are highly unbalanced, wherein most slots are inactive in the dialogue, especially under the setting of multi-domain.

In the second level, the *value matching errors* can be further divided into the *tracking errors* and the *attended to wrong slot*, and the *classification errors* consist of four sub-categories: *tracking errors*, *attended to wrong slot*, *insufficient information through gate* and *useless copy*, respectively. The first category of error is caused by the belief tracker which produces flawed context representation, while the other three are pertaining to the copy mechanism. *Attended to wrong slot* denotes that the slot attention module focuses on wrong slots and *insufficient information through gate* represents that the target of the slot attention module is correct but only little of the summarized information has been allowed to pass through the gate. Moreover, *useless copy* indicates the error that the model has presumes a flawed connection between two slots. Regarding these three errors driven by the proposed copy mechanism, the *attended to wrong slot* holds the largest proportion. It indicates that incorrectly modeling the interaction among slots constitutes the primary suffering of our proposed copy mechanism, allowing the advanced techniques to further improve its effectiveness.

VI. Conclusion and Future Work

In this work, we present a hierarchical model equipped with a well-devised copy mechanism, whcih is able to effectively model the second-order slot interactions in DST. In specific, we employ a contextual encoder with the state reuse approach for encoding, apply an attention-based belief tracker for managing dialogue states, and develop a soft copy mechanism for selectively copying possible values from other slots. Extensive experiments on the benchmark dataset have validated the superiority of our proposed method over state-of-the-art baselines.

Given that the graph neural networks (GNNs) have gained much success nowadays, we plan to apply GNNs to model high-order slot interactions (i.e., the interactions beyond the direct relationship between two slots), and justify its effectiveness in DST.

References

[1] P. Budzianowski, T. Wen, B. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gasic, "Multiwoz - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling," in *EMNLP*. ACL, 2018, pp. 5016–5026.

[2] C. Wu, A. Madotto, E. Hosseini-Asl, C. Xiong, R. Socher, and P. Fung, "Transferable multi-domain state generator for task-oriented dialogue systems," in *ACL*. ACL, 2019, pp. 808–819.

[3] J. D. Williams and S. J. Young, "Partially observable markov decision processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 393–422, 2007.

[4] M. Henderson, B. Thomson, and S. J. Young, "Word-based dialog state tracking with recurrent neural networks," in *SIGDIAL*. ACL, 2014, pp. 292–299.

[5] T. Wen, D. Vandyke, N. Mrksic, M. Gasic, L. M. Rojas-Barahona, P. Su, S. Ultes, and S. J. Young, "A network-based end-to-end trainable task-oriented dialogue system," in *EACL*. ACL, 2017, pp. 438–449.

[6] P. Xu and Q. Hu, "An end-to-end approach for handling unknown slot values in dialogue state tracking," in *ACL*. ACL, 2018, pp. 1448–1457.

[7] N. Mrksic, D. Ó. Séaghdha, T. Wen, B. Thomson, and S. J. Young, "Neural belief tracker: Data-driven dialogue state tracking," in *ACL*. ACL, 2017, pp. 1777–1788.

[8] Z. Zhang, M. Huang, Z. Zhao, F. Ji, H. Chen, and X. Zhu, "Memory-augmented dialogue management for task-oriented dialogue systems," *TOIS*, vol. 37, no. 3, pp. 34:1–34:30, 2019.

[9] H. Le, R. Socher, and S. C. Hoi, "Non-autoregressive dialog state tracking," in *ICLR*, 2020, pp. 1–21.

[10] Z. Li, Z. Li, J. Zhang, Y. Feng, and J. Zhou, "Bridging text and video: A universal multimodal transformer for audio-visual scene-aware dialog," *TASLP*, vol. 29, pp. 2476–2483, 2021.

[11] J. Zeng, Y. Yin, Y. Liu, Y. Ge, and J. Su, "Domain adaptive meta-learning for dialogue state tracking," *TASLP*, vol. 29, pp. 2493–2501, 2021.

[12] Z. Zhang, J. Li, and H. Zhao, "Multi-turn dialogue reading comprehension with pivot turns and knowledge," *TASLP*, vol. 29, pp. 1161–1173, 2021.

[13] J. Gao, M. Galley, and L. Li, "Neural approaches to conversational AI," in *SIGIR*. ACM, 2018, pp. 1371–1374.

[14] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," in *ACL*. ACL, 2019, pp. 2978–2988.

[15] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *AAAI*. AAAI, 2016, pp. 3776–3784.

[16] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio, "A hierarchical latent variable encoder-decoder model for generating dialogues," in *AAAI*, 2017, pp. 3295–3301.

[17] H. Lee, J. Lee, and T. Kim, "SUMBT: slot-utterance matching for universal and scalable belief tracking," in *ACL*. ACL, 2019, pp. 5478–5483.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[19] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*. ACL, 2019, pp. 4171–4186.

[20] E. Hosseini-Asl, B. McCann, C. Wu, S. Yavuz, and R. Socher, "A simple language model for task-oriented dialogue," in *NeurIPS*, 2020, pp. 20 179–20 191.

[21] M. Heck, C. van Niekerk, N. Lubis, C. Geishauser, H. Lin, M. Moresi, and M. Gasic, "Trippy: A triple copy strategy for value independent neural dialog state tracking," in *SIGDIAL*. ACL, 2020, pp. 35–44.

[22] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," in *ICLR*, 2017, pp. 1–13.

[23] H. Huang, C. Zhu, Y. Shen, and W. Chen, "Fusionnet: Fusing via fully-aware attention with application to machine comprehension," in *ICLR*, 2018, pp. 1–20.

[24] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," in *NeurIPS*, 2019, pp. 5754–5764.

[25] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *NAACL-HLT*. ACL, 2018, pp. 2227–2237.

[26] L. Ren, J. Ni, and J. J. McAuley, "Scalable and accurate dialogue state tracking via hierarchical sequence generation," in *EMNLP*. ACL, 2019, pp. 1876–1885.

[27] A. Kumar, P. Ku, A. K. Goyal, A. Metallinou, and D. Hakkani-Tür, "MA-DST: multi-attention-based scalable dialog state tracking," in *AAAI*. AAAI, 2020, pp. 8107–8114.

[28] S. Kim, S. Yang, G. Kim, and S. Lee, "Efficient dialogue state tracking by selectively overwriting memory," in *ACL*. ACL, 2020, pp. 567–582.

[29] L. Zhou and K. Small, "Multi-domain dialogue state tracking as dynamic knowledge graph enhanced question answering," *CoRR*, vol. abs/1911.06192, 2019.

[30] L. Chen, B. Lv, C. Wang, S. Zhu, B. Tan, and K. Yu, "Schema-guided multi-domain dialogue state tracking with graph attention neural networks," in *AAAI*. AAAI, 2020, pp. 7521–7528.

[31] J. Hu, Y. Yang, C. Chen, L. He, and Z. Yu, "SAS: dialogue state tracking via slot attention and slot information sharing," in *ACL*. ACL, 2020, pp. 6366–6375.

[32] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *NeurIPS*, 2015, pp. 2692–2700.

[33] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *ACL*. ACL, 2017, pp. 1073–1083.

[34] S. He, C. Liu, K. Liu, and J. Zhao, "Generating natural answers by incorporating copying and retrieving mechanisms in sequence-to-sequence learning," in *ACL*. ACL, 2017, pp. 199–208.

[35] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, "Extracting relational facts by an end-to-end neural model with copy mechanism," in *ACL*. ACL, 2018, pp. 506–514.

[36] M. Dehghani, S. Rothe, E. Alfonseca, and P. Fleury, "Learning to attend, copy, and generate for session-based query suggestion," in *CIKM*. ACM, 2017, pp. 1747–1756.

[37] T. Yao, Y. Pan, Y. Li, and T. Mei, "Incorporating copying mechanism in image captioning for learning novel objects," in *CVPR*. IEEE, 2017, pp. 5263–5271.

[38] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP*. ACL, 2014, pp. 1724–1734.

[39] M. Eric, R. Goel, S. Paul, A. Sethi, S. Agarwal, S. Gao, A. Kumar, A. K. Goyal, P. Ku, and D. Hakkani-Tür, "Multiwoz 2.1: A consolidated multi-domain dialogue dataset with state corrections and state tracking baselines," in *LREC*. ELRA, 2020, pp. 422–428.

[40] J. Zhang, K. Hashimoto, C. Wu, Y. Wan, P. S. Yu, R. Socher, and C. Xiong, "Find or classify? dual strategy for slot-value predictions on multi-domain dialog state tracking," *CoRR*, vol. abs/1910.03544, 2019.

[41] R. Goel, S. Paul, and D. Hakkani-Tür, "Hyst: A hybrid approach for flexible and accurate dialogue state tracking," in *ISCA*, 2019, pp. 1458–1462.

[42] S. Gao, A. Sethi, S. Agarwal, T. Chung, and D. Hakkani-Tür, "Dialog state tracking: A neural reading comprehension approach," in *SIGDIAL*. ACL, 2019, pp. 264–273.

Fangkai Jiao is a Master student in the School of Computer Science and Technology, Shandong University. He received the B.E. degree from the School of Software, Shandong University, in 2019. His research interests include pre-training, machine reading comprehension and dialogue system.

Yangyang Guo is currently a research fellow with the National University of Singapore. He has published several papers in top conferences and journals such as IEEE TIP, IEEE TMM, IEEE TKDE and ACM TOIS. He has served as a Regular Reviewer for journals, including IEEE TMM, IEEE TKDE, ACM ToMM.

Minlie Huang (Member, IEEE) received the Ph.D. degree from Tsinghua University, Beijing, China, in 2006. He is currently an Associate Professor with the Department of Computer Science and Technology, Tsinghua University. He has authored or coauthored more than 60 papers in premier conferences and journals (ACL, EMNLP, AAAI, IJCAI, WWW, SIGIR, etc.) His research interests include natural language processing, particularly in dialog systems, reading comprehension, and sentiment analysis. His work on emotional chatting machines was reported by MIT Technology Review, the Guardian, Nvidia, and many other mass media. He serves as standing reviewer for TACL, Area Chairs for ACL 2020/2016, EMNLP 2019/2014/2011, and Senior PC members for AAAI 2017–2020 and IJCAI 2017–2020, and reviewers for TASLP, TKDE, TOIS, TPAMI, etc. He is a nominee of ACL 2019 best demo papers, the recipient of IJCAI 2018 distinguished paper award, CCL 2018 best demo award, NLPCC 2015 best paper award, Hanvon Youngth Innovation Award in 2018, and Wuwenjun AI Award in 2019. He was supported by a NSFC key project, several NSFC regular projects, and many IT companies.

Liqiang Nie (Senior Member, IEEE) is currently a professor with Shandong University and the dean with the Shandong AI institute. He received his B.Eng. and Ph.D. degree from Xi'an Jiaotong University and National University of Singapore (NUS), respectively. After PhD, Dr. Nie continued his research in NUS as a research follow for three years. His research interests lie primarily in multimedia computing and information retrieval. Dr. Nie has co-/authored more than 200 papers and 4 books, received more than 11,000 Google Scholar citations as of Aug 2021. He is an AE of IEEE TKDE, IEEE TMM, ACM ToMM, and Information Science. Meanwhile, he is an area chair of ACM MM 2018-2021. He has received many awards, like ACM MM and SIGIR best paper honorable mention in 2019, SIGMM rising star in 2020, TR35 China 2020, DAMO Academy Young Fellow in 2020, and SIGIR best student paper in 2021.